

ABSTRACT OF THE DISCLOSURE

An incrementally-scalable file system and method. The system architecture enables file systems to be scaled by adding resources, such as additional filers and/or file servers, without requiring that the system be

5 taken offline or being known to client applications. The system also provides for load balancing file accesses by distributing files across the various file storage resources in the system, as dictated by the relative capacities of said storage resources. The system provides one or more

10 "virtual" file system volumes in a manner that makes it appear to client applications that all of the file system's storage space resides on the virtual volume(s), while in reality the files may be stored on many more physical volumes on the filers and/or file servers in the system. This functionality is enabled through a software "virtualization" storage abstraction layer that intercepts file system requests and remaps the

15 virtual volume location to the actual physical location of the files on the various filers and file servers in the system. This scheme is implemented through the use of two software components: 1) an "agent" software module that determines and knows how files are distributed throughout the system, and 2) a "shim" that is able to intercept file system requests.

20 For Microsoft Windows clients, the shim is implemented as a file system filter. For Unix-variant clients, the shim is implemented as one or more NFS daemons. When new storage resources are added to the file

system, files from existing storage devices are migrated to the new
resources in a manner that makes the migration appear to be "invisible" to
25 client applications, and load balancing is obtained.

DDOT-TECHS